

# Evolving Fuzzy Classifier based on Clustering Algorithm and Drift Detection for Fault Diagnosis Applications

Maurilio Inacio<sup>1</sup>, Andre Lemos<sup>2</sup>, and Walmir Caminhas<sup>3</sup>

<sup>1</sup> Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte, 31270-901, Brazil

<sup>1</sup> Dept. of Computer Engineering, Faculdade de Ciencia e Tecnologia de Montes Claros, Montes Claros, 39400-142, Brazil  
maurilio.j.inacio@gmail.com

<sup>2,3</sup> Dept. of Electronic Engineering, Federal University of Minas Gerais, Belo Horizonte, 31270-901, Brazil  
andrepaim@gmail.com  
caminhas@cpdee.ufmg.br

## ABSTRACT

Nowadays, in several areas, efficient fault diagnosis methods for complex machinery and equipments are required. Several fault diagnosis methods based on different theories and approaches have been proposed in the literature. In general, these methods use mathematical/statistical models, accumulated experience, or even process historical data to perform fault diagnosis. Although methods based on models or experience have shown to be effective, they have the disadvantage of requiring previous knowledge of the dynamic system in question. On the contrary, methods based on process historical data do not require a *prior* knowledge, they are based solely on data obtained directly from the dynamic system. The application of so-called “Evolving Intelligent Systems” to accomplish fault diagnosis from process data have been shown a promising approach. This paper proposes an evolving fuzzy classifier based on a new approach that combines a recursive clustering algorithm and a drift detection method and its application on dynamic systems fault diagnosis. The novel approach provides greater robustness to outliers and noise present in data from process sensors. The classifier is evaluated in fault diagnosis of an interacting tank system and the results are promising.

## 1. INTRODUCTION

Nowadays, the advance of technology has resulted in the emergence of machinery and complex equipments, which impose great challenges for its management and maintenance. In many industries, for instance, fault diagnosis in major processes is vitally important to assure normal operation of a

plant and avoid economic losses, security reductions and environmental damages. This context led to the emergence of new concepts on management and maintenance of machinery and equipments, such as Condition-Based Maintenance (CBM). In CBM, machine or equipment data obtained in real time are used to infer its working condition (or faulty condition), allowing maintenance scheduling and preventing equipment crashes. Another concept has emerged based on CBM, the concept of intelligent maintenance (Vachtsevanos, Lewis, Roeme, Hess, & Wu, 2006).

In past decades several fault diagnosis methods based on different approaches have been proposed in the literature. These methods use mathematical models, statistical models, accumulated experience, or process historical data to perform fault diagnosis (Venkatasubramanian, 2005). Fault diagnosis methods based on process historical data have received great emphasis recently (Abellan-Nebot & Subirón, 2010) and several works have already proposed data based diagnostics methods employing intelligent systems, mainly artificial neural networks and fuzzy systems (Jardine, Lin, & Banjevic, 2006). Nevertheless, despite the good performance achieved by intelligent systems in fault diagnosis, they tend to face difficulties when the problem involves complex non-stationary dynamic systems. In this systems, physical parameters, operating characteristics and fault behaviours change over time, requiring an adaptive fault diagnosis system, able to self-adapt to cope with changes in the monitored system. In order to address fault diagnosis in this cases, some works propose the use of so-called “Evolving Intelligent Systems” (Lughofer & Guardiola, 2008; Filev, Chinnam, Tseng, & Baruah, 2010; Lemos, Caminhas, & Gomide, 2013).

Based on artificial neural networks, fuzzy inference systems or a combination of both, the neurofuzzy networks, the evolv-

Maurilio Inacio et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ing intelligent systems are systems whose main characteristic is the ability to gradually determine both its structure and parameters from input data acquired in online mode and often in real time. Evolving intelligent systems applications has been growing in recent years. Many authors have obtained successful applications in real world complex problems involving modeling, control, classification or prediction (Angelov, Filev, & Kasabov, 2010). Evolving clustering algorithm is the most widely used approach to define the structure of an evolving intelligent system (Kasabov & Song, 2002; Angelov & Filev, 2003; Leng, McGinnity, & Prasad, 2005; Rong, Sundararajan, Huang, & Saratchandran, 2006; Lughofer, 2008; Soleimani-B., Lucas, & Araabi, 2010; Lima, Hell, Gomide, & Ballini, 2010; Lemos, Caminhas, & Gomide, 2011). These algorithms generally adopt a mechanism to update the structure (creation/modification/removal of clusters) and parameters of the system using some measure of similarity between input data samples and existing clusters. This mechanism may lead to an erroneous definition of the structure, since outliers or noisy samples (as usually are the data acquired by sensors in industrial environments) which exceeds the measure of similarity can generate clusters that do not effectively represent the data spacial structure (Lemos et al., 2011).

In fault diagnosis problems, the use of evolving intelligent systems based on recursive clustering algorithms robust to outliers and data noise is mandatory. In this case, each new cluster created is usually associated with a new faulty condition. Thus, if the clustering procedure is not robust, the fault diagnosis model tends to have a high false alarm rate, i.e., new faulty conditions are erroneously detected. Considering this context, this paper proposes a fault diagnosis approach based on an evolving fuzzy classifier which uses a new robust unsupervised recursive clustering algorithm. The unsupervised recursive clustering algorithm classifier consists of a modified version of the Gustafson-Kessel (GK) clustering algorithm (Gustafson & Kessel, 1979) with the incorporation of the Drift Detection Method (DDM) (Gama, Medas, Castillo, & Rodrigues, 2004).

Considered a powerful clustering algorithm, GK clustering algorithm unlike many others allows the identification of clusters with different shapes and orientations in space. The algorithm employs a technique to adapt the distance metric to the shape of each cluster using a estimation of the cluster covariance matrix. Furthermore, the algorithm has also the advantage of being relatively insensitive to data scale and initialization of the partition matrix (Filev & Georgieva, 2010). Drift detection, according to the literature, is a method to detect gradual changes in the context of input data. By context, it is understood as a set of generated data when the process is stationary. Drift detection methods are suitable for applications involving machine learning, where algorithms are applied to real world problems, in complex, non-stationary and dynamic environments (Sebastião & Gama, 2009). Among

several methods proposed for drift detection, the DDM algorithm employs simple and computationally efficient method to detect moments when changes occur and it can be embedded into any learning algorithm, increasing its efficiency in problems involving non-stationary dynamic models.

In this paper, a new unsupervised recursive clustering algorithm is proposed, where any clustering update depends not only on the similarity measure, but also on the monitoring changes in the input data flow, which gives the algorithm a greater robustness to the presence of outliers and noise. A merging cluster mechanism was also incorporated into the algorithm to enable the removal of redundant clusters. The fuzzy rule base of the proposed classifier is updated whenever the cluster structure is modified. The clusters centers and covariance matrices are used as parameters of fuzzy rules. Multivariate Gaussian membership functions are employed in the rules to avoid information loss when there is interaction between input variables. Regarding the characteristics of the proposed recursive clustering algorithm, the main benefits achieved by the classifier used in this work are: 1) the ability to learn the dynamic system model in online mode and, if necessary, in real time; 2) the ability to adapt whenever changes are detected in the monitored system, allowing the application to real problems; 3) low false alarm rate and high fault isolation rate due to the robustness to outliers and noise, increasing the reliability of diagnosis. To evaluate the performance of the proposed approach in fault diagnosis, an interacting tank system simulator was used to simulate normal and several faulty conditions. Outliers and noise were added to the simulated data to evaluate the robustness of the proposed algorithms.

After this introduction, the rest of the paper proceeds as follows. Section 2 presents the theoretical concepts regarding recursive clustering algorithm, drift detection method and presents the proposed recursive clustering algorithm. Next, Section 3 presents the proposed classifier and its application in fault diagnosis. Section 4 presents the simulations and results. Finally, Section 5 presents the conclusion and suggestions for future works.

## 2. THEORETICAL CONCEPTS: RECURSIVE CLUSTERING ALGORITHM AND DRIFT DETECTION

### 2.1. Recursive Gustaffson-Kessel Algorithm

Clustering algorithms are among the most useful tools to solve pattern recognition problems, where involves analysis of non-labeled data, or unsupervised learning (Duda, Hart, & Stork, 2001). Over the past decades, thousands of clustering algorithms have been proposed (Jain, 2010). GK algorithm, unlike many clustering algorithms that employ Euclidian distance as measure of similarity, employs Mahalanobis-like distance, which allows the identification of clusters with ellipsoidal shapes. In this algorithm the distance is defined as fol-

lows:

$$d_{ik}^2 = (x_k - v_i) A_i (x_k - v_i)^T \quad (1)$$

where  $d_{ik}^2$  represents the distance between an input data sample  $x_k = [x_{k1}, \dots, x_{kn}]$ ,  $k = 1, \dots, N$ , and the cluster center  $v_i$ ,  $i = 1, \dots, c$ , where  $N$  is the number of data samples,  $n$  is the number of data dimensions, and  $c$  is the number of clusters. The norm-inducing matrix  $A_i$ ,  $i = 1, \dots, c$ , defines the shape and orientation of each cluster in space. An iterative process is used in the GK algorithm to estimate the parameters of the clusters (the cluster center and fuzzy covariance matrix). This process is finished when a certain convergence criterion is reached. An extended version of the GK algorithm named evolving GK-like algorithm (eGKL) is proposed in Filev and Georgieva (2010). This approach estimates the number of clusters and performs the adaptation of its parameters recursively, maintaining the advantages of the GK algorithm. To evaluate the similarity between a new sample data and one of the existing clusters, the eGKL algorithm employs the Mahalanobis distance, defined as follows:

$$D_{ik}^2 = (x_k - v_i) F_i^{-1} (x_k - v_i)^T \quad (2)$$

where  $F_i$ ,  $i = 1, \dots, c$  is a covariance matrix. Thus, the current data sample belongs to an existing cluster if the distance to the cluster center is smaller than the cluster radius. The eGKL algorithm uses an approach inspired in concepts of statistical process control to estimate the radius of each cluster. In this approach, it is assumed that a sample belongs to a cluster if the following relationship holds:

$$D_{ik}^2 < \chi_{n,\beta}^2 \quad (3)$$

where  $\chi_{n,\beta}^2$  is the value of a Chi-squared distribution,  $n$  is the degrees of freedom and  $\beta$  is the confidence interval. The degrees of freedom  $n$  correspond to the input space dimension and confidence interval  $\beta$  is a parameter of the algorithm. This approach has the advantage of avoiding the problem called “curse of dimensionality” (Hastie, Tibshirani, & Friedman, n.d.), i.e., the problem of increasing the distance between two adjacent points with the increase in the input space dimensionality, since  $\chi_{n,\beta}^2$  is proportional to the dimension of the input data. If the condition given by Eq. (3) is satisfied, it means that the current data sample belongs to a cluster, so the cluster parameters are updated. Otherwise, it is assumed that the current data sample does not belong to any one of the existing clusters, and a new cluster is created. The complete procedures of the eGKL algorithm can be seen in Filev and Georgieva (2010).

## 2.2. Drift Detection Method

In the literature, several drift detection methods have been proposed. In general, they can be classified into two categories: methods that perform adaptive learning at regular intervals regardless of the occurrence of changes, and meth-

ods that detect changes first and subsequently adapt the learning to these changes (Sebastião & Gama, 2009). Belonging to the second category, the DDM algorithm employs a simple method with direct application. This method is based on monitoring the number of errors produced by a learning model during prediction. The method uses the Binomial distribution to determine the general form of the probability for the random variable that represents the number of prediction errors into a sequence of  $n$  input data samples. In DDM algorithm, for each  $k$  data sample sequences, the error rate is the probability of the prediction error  $p_k$  with standard deviation  $s_k = \sqrt{p_k(1-p_k)/k}$ . According to the Probability Approximately Correct (PAC) learning model (Mitchell, 1997), the error rate of the learning algorithm decreases with the increase of input data samples, and if the distribution is stationary, a significant increase in the error rate suggests context changes. In this case, it is assumed that the current model is inappropriate and should be updated. In DDM algorithm, while monitoring the error, it defines a warning and a drift level. When  $p_k + s_k$  exceeds the warning level, the data samples are stored in memory. However, if  $p_k + s_k$  exceeds the drift level, it is considered that there is a context change. In this situation, the model induced by the learning algorithm should be updated with the data samples stored since the time that the warning level has been reached. It is possible that the error increases and, after reaching the warning level, it decreases to lower levels. This situation corresponds to a false alarm, where there is no change of context and, therefore, no action is required and the data samples stored in the memory are no longer needed. More details about the DDM method can be found in Gama et al. (2004).

## 2.3. Proposed Recursive Clustering Algorithm

The algorithm proposed in this work consists of an unsupervised recursive clustering algorithm with a new mechanism of clustering update. The algorithm is a recursive version of the GK algorithm, inspired by the eGKL algorithm, and incorporating the DDM algorithm. Thus, clustering is performed in online mode and, if necessary, in real time.

Considering that there is no a priori information about the clustering structure neither a initial set of input data samples, the proposed algorithm starts by associating the center of the first cluster  $c_1$  to the first input data sample  $x_1$ . The corresponding covariance matrix  $F_1$ , the learning rate  $\alpha_1$  and the number of samples associated with the first cluster  $M_1$  are defined as follows:  $c_1 = x_1$ ;  $F_1 = F_{init}$ ;  $\alpha_1 = \alpha_{init}$ ;  $M_1 = 1$ , where  $F_{init} = \gamma I$ ;  $I$  is an identity matrix of  $n$  size;  $\gamma$  is a small positive number (default value:  $\gamma = 10^{-2}$ ) and  $\alpha_{init} \in [0, 1]$  is the initial learning rate (default value:  $\alpha_{init} = 0.5$ ). If all data samples are processed, the algorithm stops, otherwise, a new input data sample  $x_k$  is obtained and the distance between the data sample and the

centers of the existing clusters is computed as:

$$D_{ik}^2 = (x_k - v_i)F_i^{-1}(x_k - v_i)^T \quad (4)$$

The similarity between the current data sample and the existing clusters is verified by the similarity condition:

$$D_{ik}^2 < \chi_{n,\beta}^2 \quad (5)$$

where  $\chi_{n,\beta}^2$  is the value of a Chi-squared distribution,  $n$  is the degrees of freedom and  $\beta$  is the confidence interval. The degrees of freedom  $n$  correspond to the input space dimension and confidence interval  $\beta$  is a parameter of the algorithm. If similarity condition given by Eq. (5) is met for a cluster, it is assumed that the current sample belongs to this cluster. The cluster parameters (center, covariance matrix, learning rate and number of samples in the cluster) are then updated as follows:

$$v_q = v_q + \alpha_q(x_k - v_q) \quad (6)$$

$$F_q = F_q + \alpha_q((x_k - v_q)^T(x_k - v_q) - F_q) \quad (7)$$

$$\alpha_q = \frac{\alpha_{init}}{M_q} \quad (8)$$

$$M_q = M_q + 1 \quad (9)$$

where  $q = \arg \min_{i=1,\dots,c} (D_{ik}^2)$ . If the similarity condition given by

Eq. (5) is not met, it is assumed that the current sample does not belong to any existing cluster. Then, the algorithm increments a variable that represents the number of dissimilarities,  $M_{dis} = M_{dis} + 1$ , then the error probability and standard deviation are computed as follows:

$$p = \frac{M_{dis}}{k} \quad (10)$$

$$s = \sqrt{p(1-p)/k} \quad (11)$$

In this algorithm, the  $p$  and  $s$  values are stored whenever  $p+s$  reach the lowest value during the process, obtaining  $p_{min}$  and  $s_{min}$ . If the following condition is met:

$$p + s < p_{min} + s_{min} \quad (12)$$

then  $p_{min} = p$  and  $s_{min} = s$ . Note that, when algorithm starts, the  $p$  and  $s$  values must be initialized as a positive number, it is suggested set as one for each value. To decide whether the current data sample  $x_k$  represents a new cluster or it is just an outlier, warning and drift conditions are evaluated. The warning condition is verified as:

$$p + s > p_{min} + z_1 \cdot s_{min} \quad (13)$$

where  $z_1$  is the warning level (default value:  $z_1 = 2$ ). If the warning level is reached, then the current data sample is stored in a window of samples  $W(data)_j$ ,  $j = 1, \dots, m$  (where  $m$  is the current size of the window) and then, the drift condition is evaluated. Otherwise, the algorithm processes the next input data sample. Drift condition is verified

as:

$$p + s > p_{min} + z_2 \cdot s_{min} \quad (14)$$

where  $z_2$  is the drift level (default value:  $z_2 = 3$ ). If the drift level is reached, a new cluster is created,  $c = c + 1$ , and the center and the covariance matrix of the new cluster are determined by the samples stored in the data window as follows:

$$v_c = \frac{1}{m} \sum_{j=1}^m W(data)_j \quad (15)$$

$$F_c = cov(W(data)_j) \quad (16)$$

The remaining parameters of the new cluster (learning rate and number of samples in the cluster) are initialized as:  $\alpha_c = \alpha_{init}$ ;  $M_c = 1$ .

In order to avoid redundant cluster formation, during the update, the similarity between clusters is checked. To that end, distances between the centers of the clusters are computed as follows:

$$D_{ij}^2 = (v_i - v_j)F_i^{-1}(v_i - v_j)^T \quad (17)$$

$$D_{ji}^2 = (v_j - v_i)F_j^{-1}(v_j - v_i)^T \quad (18)$$

If one of the following similarity conditions is met for two existing clusters  $i$  and  $j$ ,

$$D_{ij}^2 < \chi_{n,\beta}^2 \quad (19)$$

$$D_{ji}^2 < \chi_{n,\beta}^2 \quad (20)$$

the clusters are merged. These clusters have a hyper ellipsoidal shape, defined by a mean vector, a covariance matrix, and a number of samples associated with each one. The combination of these two clusters produce a new one with parameters computed as follows (Kelly, 1994):

$$M_i = M_i + M_j \quad (21)$$

$$v_i = \frac{M_i}{M_i + M_j} v_i + \frac{M_j}{M_i + M_j} v_j \quad (22)$$

$$F_i = \frac{M_i - 1}{M_i + M_j + 1} F_i + \frac{M_j - 1}{M_i + M_j + 1} F_j + \frac{M_i M_j}{M_i + M_j(M_i + M_j - 1)} (v_i - v_j)^T (v_i - v_j) \quad (23)$$

Algorithm 1 summarizes the proposed recursive clustering algorithm.

### 3. PROPOSED EVOLVING FUZZY CLASSIFIER FOR FAULT DIAGNOSIS

In many current applications, the use of algorithms for pattern classification is present, such as fingerprint recognition for security systems, handwriting recognition on touch screen computers, DNA sequences identification in medical diagnostic softwares and fault diagnosis in industrial equipments. In this

**Algorithm 1:** Recursive Clustering Algorithm with Drift Detection

---

**Input:**  $x_k$ ,  $\chi_{n,\beta}^2$ ,  $F_{init}$ ,  $\alpha_{init}$ ,  $z_1$ ,  $z_2$ ;

**Output:**  $v_i$ ,  $F_i$ ;

Read the first data sample  $x_1$ ;

Initialize the first cluster;

**for**  $k = 2, 3, \dots$  **do**

- Read  $x_k$ ;
- Compute  $D_{ik}^2$  for all clusters;
- Identify the closest cluster;
- if**  $D_{ik}^2 < \chi_{n,\beta}^2$  **then**

  - | Update the closest cluster;

- else**

  - | Update the dissimilarity number  $M_{dis}$ ;
  - | Compute  $p$  and  $s$ ;
  - | **if**  $p + s < p_{min} + s_{min}$  **then**

    - | | Update  $p_{min}$  and  $s_{min}$ ;

  - | **end if**
  - | **if**  $p + s > p_{min} + z_1 \cdot s_{min}$  **then**

    - | | Store  $x_k$  in the data window  $W(data)_j$ ;

  - | **end if**
  - | **if**  $p + s > p_{min} + z_2 \cdot s_{min}$  **then**

    - | | Create new cluster;

  - | **end if**

- end if**
- Compute  $D_{ij}^2$  and  $D_{ji}^2$  for all clusters;
- if**  $D_{ij}^2 < \chi_{n,\beta}^2$  or  $D_{ji}^2 < \chi_{n,\beta}^2$  **then**

  - | Merge redundant clusters;

- end if**

**end for**

---

context, the problem of pattern classification consists in assigning a class or a category for each data sample from a set of “raw” data (Duda et al., 2001). Pattern classification algorithms based on fuzzy rules have been used in many applications due to their advantages in relation to classic algorithms for pattern classification, especially by the good prediction performance in real problems and good transparency in linguistic rules (Jang, Sun, & Mizutani, 1997), which allows an easy comprehension of the dependence between pattern characteristics. The typical architecture of a fuzzy classifier consists of a set of IF ... THEN fuzzy rules, defined as:

$$\text{RULE}_i : \text{IF } x_1 \text{ IS } \mu_{i1} \text{ AND } \dots \text{ AND } x_n \text{ IS } \mu_{in} \text{ THEN } y_i = L_i \quad (24)$$

where  $[x_{k1}, \dots, x_{kn}]$  are the input variables or patterns of  $n$  dimensionality;  $[\mu_{i1}, \dots, \mu_{in}]$  are antecedent fuzzy sets of the  $i$ th fuzzy rule;  $y_i$  is the output;  $L_i$  is the crisp output corresponding to the class label from the set  $[1, \dots, K]$ , where  $K$  is the number of classes. For each new input data sample  $x_k$ , the classification is obtained by assigning to it the label of the class associated with the rule having the highest activation degree. The class is determined as follows:

$$y_i = L_{i^*} \quad (25)$$

where  $i^* = \arg \max_{1 \leq i \leq R} (\tau_i)$ ;  $R$  is the number of fuzzy rules and  $\tau_i$  is the activation degree of the  $i$ th fuzzy rule, defined by a t-norm, usually expressed as a product operator:

$$\tau_i = \prod_{j=1}^n \mu_{ij}(x_j) \quad (26)$$

where  $\mu_{ij}$  are the membership functions of fuzzy sets defined by Gaussians:

$$\mu_{ij} = e^{-\frac{1}{2} \left( \frac{(x_j - v_{ij})^2}{\sigma_{ij}^2} \right)} \quad (27)$$

where  $v_{ij}$  and  $\sigma_{ij}^2$  represent respectively the membership functions center and variance. Usually, to implement this fuzzy classifier architecture, clustering is performed in the input and/or output space. Then, rules are created using one-dimensional (or univariate) fuzzy sets, generated from the projection of the clusters in the axis of each variable. According to Lemos et al. (2011), this approach can lead to information loss if there is interaction between variables, and to avoid this, the authors propose the use of multivariate Gaussian membership functions to represent antecedent fuzzy sets of each rule. These membership functions are described as:

$$H(x) = e^{-\frac{1}{2} ((x - v)\Sigma^{-1}(x - v)^T)} \quad (28)$$

where  $v$  is a  $1 \times n$  central vector and  $\Sigma$  is a  $n \times n$  symmetric positive definite matrix. The central vector is defined as the modal value and represents  $H(x)$  typical value and the  $\Sigma$  matrix denotes the dispersion and represents  $H(x)$  spreading. Thus, each cluster found by the clustering algorithm is associated with a fuzzy rule and the multivariate Gaussian membership function parameters are defined as the parameters of the corresponding cluster. If multivariate Gaussian membership functions are used, the fuzzy classifier will have a rule set defined as:

$$\text{RULE}_i : \text{IF } x_k \text{ IS } A_i \text{ THEN } y_i = L_i \quad (29)$$

where  $A_i$  is the fuzzy set with multivariate Gaussian membership function of the  $i$ th fuzzy rule, with parameters extracted from the corresponding cluster. In general, more than one rule can be used to describe a class, e.g, the class can be multimodal. In this case, only one rule cannot be sufficient to describe all possible variations of the same class. Thus, the fuzzy classifier aggregates rules outputs associated with the same class using a s-norm. The result of the aggregation can be interpreted like rules as follows:

$$(\text{IF } x_k \text{ IS } A_i) \text{ OR } (\text{IF } x_k \text{ IS } A_j) \text{ OR } \dots (\text{IF } x_k \text{ IS } A_k) \text{ THEN } y_i = L_i \quad (30)$$

This aggregation results in the degree of relevance of each known class. The classification of each new sample  $x_k$  is defined by the class with the highest relevance degree.

Data samples classes are not known *a priori* in some pattern classification applications. In these situations it is required

the use of an unsupervised learning process for classifier implementation. Moreover, in applications where the pattern classification should be performed in real time, the learning should be performed using incremental algorithms, processing each data sample once as a data stream. To solve these problems, the solution is to use a recursive clustering algorithm. We propose in this paper an evolving fuzzy classifier based on recursive clustering algorithm with drift detection presented in Section 2.3, which allows the creation of a fuzzy rule base in online mode and, if necessary, in real time from input data samples. This approach is different from the ones employed in traditional fuzzy classifiers, which require some training (usually supervised) conducted in off-line mode. For rule base update, the proposed evolving fuzzy classifier uses the output of the recursive clustering algorithm described in the previous section. For each new input data sample, if a new cluster is created, a new fuzzy rule given by Eq. (29) is added to the rule base, where the cluster parameters are used as parameters of the multivariable Gaussian membership function of the antecedents. The rule consequent (the crisp output corresponding to the class label) must be defined by experts or system operators, since in unsupervised learning processes incoming online samples usually are not pre-labelled. If a cluster is updated, the corresponding class label is determined as the consequent of the fuzzy rule with the highest activation degree, and the user intervention is not necessary. If two clusters are merged by the recursive clustering algorithm, the corresponding fuzzy rules are also merged to represent an unique class. It should be noted that, both the number of rules and the number of classes are determined during the evolving process, and it is not necessary to set these parameters *a priori*. Algorithm 2 summarizes the procedures of the classifier.

---

**Algorithm 2:** Evolving Fuzzy Classifier
 

---

```

Input:  $x_k$ ;
Output:  $y_k$ ;
Initialize the classifier;
for  $k = 1, 2, \dots$  do
  Read  $x_k$ ;
  Execute the recursive clustering algorithm with drift
  detection;
  if new cluster is created then
    Create new fuzzy rule;
    Define the new class elicited by expert / system
    operator;
     $y_k$  = label of the new class;
  end if
  if cluster is updated then
    Update the corresponding fuzzy rule;
    Find the most active rule;
     $y_k$  = label of the most active rule;
  end if
  if clusters are merged then
    Merge the corresponding fuzzy rules;
  end if
end for
  
```

---

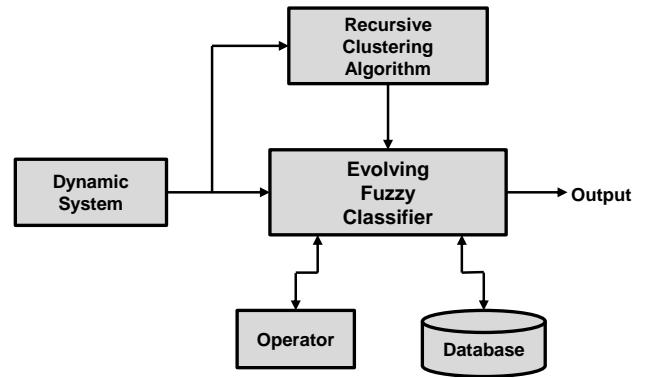


Figure 1. Fault diagnosis with the evolving fuzzy classifier.

Figure 1 illustrates the application of the proposed classifier for fault diagnosis. Data samples are obtained from a dynamic system in a continuous stream, usually provided by sensors that monitor the process. These data might require the use of pre-processing techniques for feature extraction. The rule set of the classifier starts empty at the beginning. Rules are created as the recursive clustering algorithm creates clusters to represent the data stream. Each rule will be related to a class, and each class will be related to a dynamic system condition, representing a normal operation or a fault. When a new rule is created, the system operator is notified and informs the label of the new class that defines it as a normal operation condition or as a specific fault. All of the necessary diagnostic information, the fuzzy rules and classes label, are stored in a unified database and updated while the system is used. The classifier database will contain a set of fuzzy rules and classes labels defined after an initial period of operation. When a new data sample is associated with an existing cluster, the classifier updates the corresponding fuzzy rule and classifies the dynamic system condition as the label present in the consequent of the fuzzy rule with the highest activation degree. It should be noted that, in this situation, user intervention is not required, and the classification of the dynamic system condition is performed automatically. The main feature of the classifier proposed in this work is ability to diagnose faults in a complex non-stationary dynamic system in online mode and, if necessary, in real time. The classifier does not require any *a priori* information about the dynamic model neither process historical data. This allows the classifier to construct a rule base in an evolving way and, with the aid of the operator, to learn to diagnose faults as they occur. Thus, the proposed classifier is able to adapt to the dynamic system, making it possible to diagnose faults not previously known.

#### 4. SIMULATIONS AND RESULTS

The proposed classifier was evaluated for fault diagnosis in an interacting tank system. The interacting tank system model

employed in this work was based in the system proposed by Braga, Jota, Polito, and Pena (1995) and allows to simulate faults that resembles the faults of real industrial plants. As illustrated in Fig. 2, the system comprises of a reservoir (TQ-1) and two passively interconnected tanks (TQ-2 and TQ-3). Using the interacting tank system model is possible to perform fault simulation on the actuators (pneumatic valves and pumps), at the system components (connection pipes between tanks) and on the sensors, with different sets of parameters. The types of faults are detailed in Table 1. In the fault simulation, the system starts at normal operation, and a fault is set at half of the simulation interval. Figure 3 shows as an example the curves of the TQ-2 level, TQ-3 level, TQ-2 input flow rate and TQ-3 output flow rate in fault simulation (FCV-1 valve tightness). At the beginning of each simulation, the system is working under normal operation, and the fault starts at the half of the period.

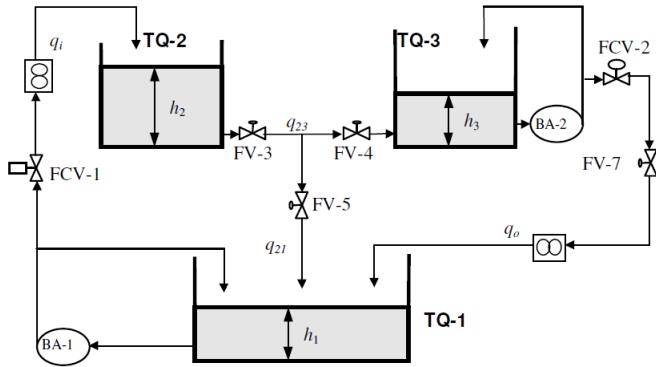


Figure 2. Representation of the interacting tank system.

Table 1. Types of faults on interacting tank system.

Index	Description
0	Normal operation
1	FCV-1 valve tightness
2	FCV-2 valve tightness
3	BA-1 pump shutdown
4	BA-2 pump shutdown
5	pipe clogging between TQ-1 and TQ-2
6	pipe clogging between TQ-1 and TQ-3
7	pipe clogging between TQ-2 and TQ-3
8	pipe leakage between TQ-2 and TQ-3
9	TQ-3 level sensor fault
10	TQ-3 output flow rate sensor fault
11	TQ-2 input flow rate sensor fault

Different scenarios were used in the fault diagnosis experiments. Each scenario consists in the simulation of sequences from 3 to 11 randomly selected fault types within a set of

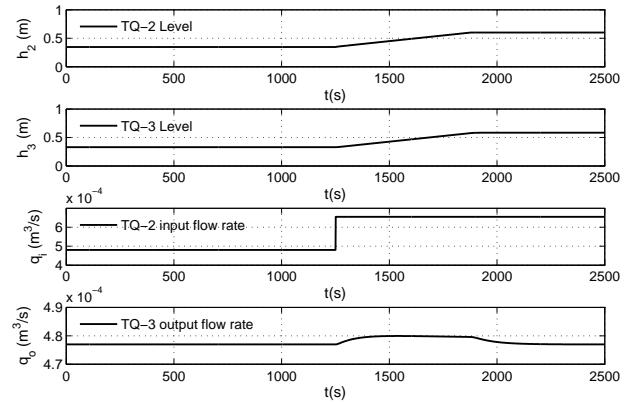


Figure 3. Fault Simulation: FCV-1 valve tightness.

faults with periods of normal operation between faults. In order to assess the robustness of the proposed classifier to the presence of noise in the data, for each monitored variable random Gaussian noise was added with a zero mean and standard deviation equal to 1% of the variable nominal value, considering normal operation of the system. As inputs of the classifier were provided in an online mode data samples related to monitored variables of the interacting tank system: TQ-2 level, TQ-3 level, TQ-2 input flow rate and TQ-3 output flow rate. For each fault sequence, the output classifier was compared to the sequence provided. Whereas the classifier starts with no fuzzy rule set, the first samples of data should match the normal operation of the system, i.e., the first rule created to describe the normal operation. For the experiments, the parameters of the recursive clustering algorithm were defined as:  $\chi_{n,\beta}^2 = 9.4877$ ;  $F_{init} = 10^{-2}I$ ;  $\alpha_{init} = 0.5$ ;  $z_1 = 2$ ;  $z_2 = 3$ .

Figure 4 show as an example the results of fault diagnosis in 5 faults scenario simulated scenario, where we can compare the estimated output (classified faults sequence) of the proposed classifier with the desired output (selected faults sequence) from input data samples. Results show that the classifier was able to correctly diagnose all the interacting tank system faults. Whereas the presence of noise in the data samples, the occurrence of false alarms or misclassification (represented by isolated points on the graph) is low, even in the scenario with the highest number of possible faults.

The classifier performance evaluation in this work was held in terms of faults detection and fault classification, as suggested in Vachtsevanos et al. (2006). Three metrics were calculated in fault detection evaluation: Probability of Detection (POD), Probability of False Alarm (POFA) and Accuracy (ACC). Regarding fault classification evaluation, the metric Fault Isolation Rate (FIR) was used. Other metrics that were used to assess the performance of the proposed classifier are: Detection

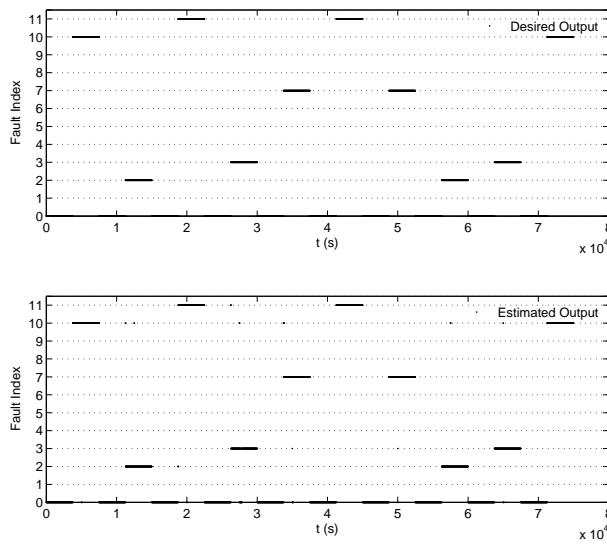


Figure 4. Desired output and estimated output by proposed classifier in 5 faults scenario.

Delay Time (DDT), Isolation Delay Time (IDT) and Operator Intervention Rate (OIR). All results of fault diagnosis experiments with interacting tank system obtained by classifier proposed in this work were compared to the results obtained using the evolving fuzzy classifier proposed by Lemos et al. (2013). For the experiments, the parameters of this alternative classifier were set to:  $w = 100$ ,  $\lambda = 0.001$ ,  $\alpha = 0.01$ ,  $T_{\mu_y} = 0.01$ . According to authors, this combination has been found experimentally to provide a good balance between the false alarm rate and the sensibility of the fault detection and diagnostic approach.

Table 2 summarizes the results for both classifiers using the fault detection metrics described. The results show that the classifier proposed in this work has higher levels of fault detection rates and accuracy in all scenarios, and no occurrence of false alarm. These results prove the efficiency of the algorithm in detecting simulated faults in the interacting tank system. Despite its lower fault detection rates and lower accuracy, the classifier proposed by Lemos et al. (2013) also not showed any false alarms.

Table 3 summarizes the results for both classifiers using the faults classification metrics described. The results show that the classifier proposed in this work presented higher fault isolation rate in all scenarios. In all scenarios the operator intervention on faults classification was very low. These results shows the ability of the classifier to automatically diagnose almost all faults after the first occurrence, and it also reveals their ability to learn. Note that, in general, the classifier proposed by (Lemos et al., 2013) had a lower performance in

faults classification than the proposed classifier and it needed more operator interventions.

Table 4 summarizes the results for both classifiers using the time metrics in fault detection and classification. A comparison between the average values for fault detection time and fault isolation time demonstrates that faults classification is faster after the first occurrence of each type of fault, since the classifier database already has the fuzzy rules and labels for all types of detected faults, not requiring an operator intervention. The results of the experiments with the classifier proposed by Lemos et al. (2013) demonstrated a faster response than the classifier proposed in this work, which is related to different update mechanisms in the clustering algorithms used in each one of the classifiers.

Table 2. Faults detection performance.

Scenario	Proposed		
	POD (%)	POFA (%)	ACC (%)
3 faults	99.38	0.00	99.67
5 faults	99.25	0.00	99.63
7 faults	99.53	0.00	99.67
9 faults	99.12	0.00	99.56
11 faults	99.20	0.00	99.60
Scenario	Lemos et al. (2013)		
	POD (%)	POFA (%)	ACC (%)
3 faults	89.35	0.00	94.67
5 faults	83.04	0.00	91.75
7 faults	82.27	0.00	91.10
9 faults	79.78	0.00	89.89
11 faults	76.02	0.00	88.01

Table 3. Faults classification performance.

Scenario	Proposed		Lemos et al. (2013)	
	FIR (%)	OIR (%)	FIR (%)	OIR (%)
3 faults	99.55	0.05	94.67	0.28
5 faults	96.76	0.04	91.88	0.29
7 faults	94.24	0.03	90.30	0.30
9 faults	92.69	0.03	89.86	0.31
11 faults	91.43	0.03	88.01	0.31

Table 4. Fault detection and classification time.

Scenario	Proposed		Lemos et al. (2013)	
	DDT (s)	IDT (s)	DDT (s)	IDT (s)
3 faults	0.065	0.003	0.015	0.003
5 faults	0.753	0.680	0.017	0.003
7 faults	1.482	1.321	0.021	0.004
9 faults	1.936	1.826	0.018	0.004
11 faults	2.327	2.204	0.018	0.004

To evaluate the robustness of the proposed classifier in the presence of outliers in the data, another experiment was conducted. In this experiment, a 5 faults scenario was simulated. Outliers were inserted in the data samples, i.e., some samples were corrupted with high variance noise. Even in the

presence of outliers, the fault diagnosis results for this experiment shows that the proposed classifier was able to correctly detect and diagnose all faults considered. This result shows that the classifier was able to correctly distinguish between outliers and valid data samples. The results of this experiment are presented in Table 5 and Table 6. Analysing these tables, one can note that the proposed classifier has virtually the same performance in fault diagnosis with absence or presence of outliers, and also not showed occurrence of false alarm. This experiment showed the greater robustness of the classifier proposed in this work when compared with the classifier proposed by Lemos et al. (2013), since the latter showed major differences in fault detection and fault classification rates in scenarios with and without outliers.

Table 5. Faults detection performance with outliers.

Scenario	Proposed		
	POD (%)	POFA (%)	ACC (%)
without outliers	99.25	0.00	99.63
with outliers	99.26	0.00	99.63
Scenario			
Lemos et al. (2013)			
Scenario	POD (%)	POFA (%)	ACC (%)
	83.78	0.00	91.75
without outliers	79.00	0.00	89.51
with outliers			

Table 6. Fault classification performance with outliers.

Scenario	Proposed		Lemos et al. (2013)	
	FIR (%)	OIR (%)	FIR (%)	OIR (%)
without outliers	96.73	0.04	91.88	0.30
with outliers	96.34	0.04	89.00	0.32

## 5. CONCLUSION

An evolving fuzzy classifier for fault diagnosis of dynamic systems was presented in this work. The proposed classifier is composed by a set of fuzzy rules created and updated based on recursive clustering algorithm. A new mechanism for cluster updating based on a drift detection method is employed, where the update of the cluster depends not only of the similarity measure, but also on the data context monitoring. As suggested by the simulation results, this feature gives the proposed classifier robustness to outliers and noise. An interacting tank system model was used for evaluation of the classifier proposed in this work. The classifier was able to detect and classify all faults with a high performance, even in the presence of outliers and noise. The high fault isolation rate and low false alarm rate obtained in all simulated scenarios showed that the recursive clustering algorithm with drift detection method was able to efficiently distinguish data samples representing clusters of invalid data. Moreover, the proposed classifier was able to automatically diagnose almost all faults, requiring operator intervention on a small percent-

age of cases. This demonstrates the advantage of the continuous and incremental learning of the classifier over other classifiers that require retraining whenever an unknown type of fault is found. The classifier proposed in this work has as advantages: the ability to learn from faults in online mode and in real time; the ability to adapt to cope with changes in the dynamic system; and robustness to the presence of outliers and noise in the input data. Summarizing, the proposed classifier has showed to be a promising alternative for application in fault diagnosis where other methods prove to be inefficient or less advantageous, because of the characteristics of such systems. In a future work, we will investigate the application of the proposed algorithm in the real time fault diagnosis and prognosis of industrial machines and equipments.

## REFERENCES

- Abellan-Nebot, J. V., & Subirón, F. R. (2010). A review of machining monitoring systems based on artificial intelligence process models. *The International Journal of Advanced Manufacturing Technology*, 47, 237-257. doi: 10.1007/s00170-009-2191-8
- Angelov, P., & Filev, D. (2003). On-line Design of Takagi-Sugeno Models. In T. Bilgiç, B. D. Baets, & O. Kaynak (Eds.), *Fuzzy Sets and Systems — IFSA 2003* (Vol. 2715, p. 576-584). Springer Berlin Heidelberg. doi: 10.1007/3-540-44967-1-69
- Angelov, P., Filev, D., & Kasabov, N. (2010). *Evolving Intelligent Systems: Methodology and Applications*. New York, USA: John Wiley & Sons.
- Braga, A. R., Jota, F. G., Polito, C. M., & Pena, R. T. (1995). Development of an interacting tank system for the study of advanced process control strategies. In *Proceedings of the 38th Midwest Symposium on Circuits and Systems* (Vol. 1, p. 441-444). doi: 10.1109/MWS-CAS.1995.504471
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. New York, USA: John Wiley & Sons.
- Filev, D., Chinnam, R. B., Tseng, F., & Baruah, P. (2010). An Industrial Strength Novelty Detection Framework for Autonomous Equipment Monitoring and Diagnostics. *IEEE Transactions on Industrial Informatics*, 6(4), 767-779. doi: 10.1109/TII.2010.2060732
- Filev, D., & Georgieva, O. (2010). An extended version of the Gustafson-Kessel algorithm for evolving data stream clustering. In P. Angelov, D. Filev, & N. Kasabov (Eds.), *Evolving Intelligent Systems: Methodology and Applications* (p. 273-300). New York, USA: John Wiley & Sons.
- Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In A. L. C. Bazzan & S. Labidi (Eds.), *Advances in Artificial Intelligence – SBIA 2004* (p. 286-295). Springer Berlin Heidelberg. doi: 10.1007/978-3-540-28645-5-29

- Gustafson, D. E., & Kessel, W. C. (1979). Fuzzy clustering with fuzzy covariance. In *Proceedings of IEEE Conference on Decision and Control* (p. 761-766).
- Hastie, T., Tibshirani, R., & Friedman, J. (n.d.). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York, USA: Springer-Verlag.
- Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), 651–666. doi: 10.1016/j.patrec.2009.09.011
- Jang, J. S. R., Sun, C. T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence*. Upper Saddle River, USA: Prentice-Hall, Inc.
- Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7), 1483–1510. doi: 10.1016/j.ymssp.2005.09.012
- Kasabov, N., & Song, Q. (2002). DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time Series Prediction. *IEEE Transactions on Fuzzy Systems*, 10(2), 144-154. doi: 10.1109/91.995117
- Kelly, P. M. (1994). *An algorithm for merging hyperellipsoidal clusters* (Tech. Rep.). LA-UR-94-3306, Los Alamos National Laboratory, Los Alamos, NM.
- Lemos, A., Caminhas, W., & Gomide, F. (2011). Multi-variable Gaussian Evolving Fuzzy Modeling System. *IEEE Transactions on Fuzzy Systems*, 19(1), 91-104. doi: 10.1109/TFUZZ.2010.2087381
- Lemos, A., Caminhas, W., & Gomide, F. (2013). Adaptive Fault Detection and Diagnosis Using an Evolving Fuzzy Classifier. *Information Sciences*, 220, 64-85. doi: DOI: 10.1016/j.ins.2011.08.030
- Leng, G., McGinnity, T. M., & Prasad, G. (2005). An Approach for On-Line Extraction of Fuzzy Rules Using a Self-Organising Fuzzy Neural Network. *Fuzzy Sets & Systems*, 150(2), 211–243. doi: DOI: 10.1016/j.fss.2004.03.001
- Lima, E., Hell, M., Gomide, F., & Ballini, R. (2010). Evolving fuzzy modeling using participatory learning. In P. Angelov, D. Filev, & N. Kasabov (Eds.), *Evolving Intelligent Systems: Methodology and Applications* (p. 67-87). New York, USA: John Wiley & Sons.
- Lughofer, E. (2008). FLEXFIS: A robust Incremental Learning Approach for Evolving Takagi–Sugeno Fuzzy Models. *IEEE Transactions on Fuzzy Systems*, 16(6), 1393-1410. doi: DOI: 10.1016/j.fss.2004.03.001
- Lughofer, E., & Guardiola, C. (2008). Applying Evolving Fuzzy Models with Adaptive Local Error Bars to On-line Fault Detection. In *Proceedings of 3rd International Workshop on Genetic and Evolving Fuzzy Systems - GEFS 2008* (p. 35-40). doi: 10.1109/GEFS.2008.4484564
- Mitchell, T. M. (1997). *Machine Learning*. New York, USA: McGraw-Hill.
- Rong, H. J., Sundararajan, N., Huang, G. B., & Saratchandran, P. (2006). Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction. *Fuzzy Sets & Systems*, 157(9), 1260-1275. doi: 10.1016/j.fss.2005.12.011
- Sebastião, R., & Gama, J. (2009). A Study on Change Detection Methods. In *Proceedings of 4th Portuguese Conference on Artificial Intelligence* (p. 353-364).
- Soleimani-B., H., Lucas, C., & Araabi, B. N. (2010). Recursive Gath-Geva clustering as a basis for evolving neuro-fuzzy modeling. *Evolving Systems*, 1(1), 59-71. doi: 10.1007/s12530-010-9006-x
- Vachtsevanos, G., Lewis, F., Roeme, M., Hess, A., & Wu, B. (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Hoboken, USA: John Wiley & Sons.
- Venkatasubramanian, V. (2005). Prognostic and diagnostic monitoring of complex systems for product lifecycle management: Challenges and opportunities. *Computers and Chemical Engineering*, 29(6), 1253-1263. doi: 10.1016/j.compchemeng.2005.02.026

## BIOGRAPHIES

**Maurilio Inacio** received the B.Sc. degree in computing engineering from the Faculdade de Ciencia e Tecnologia de Montes Claros, Montes Claros, Brazil, in 2007, and the M.Sc. degree in electrical engineering from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 2010, where he is currently working toward the Ph.D. degree with the Department of Electrical Engineering. His current research interests include evolving intelligent systems, fault diagnosis and prognosis.

**Andre Lemos** received the B.Sc. degree in computer science from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 2003, the M.Sc. degree and the Ph.D. degree in electrical engineering from the same university in 2007 and 2011, respectively. He is currently an Associate Professor with the Department of Electronic Engineering, Federal University of Minas Gerais. His current research interests include adaptive and evolving intelligent systems, machine learning and applications in time-series forecasting, fault detection and diagnosis, and nonlinear systems modeling.

**Walmir Caminhas** received the B.Sc. and M.Sc. degrees in electrical engineering from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 1987 and 1989, respectively, and the Ph.D. degree from the University of Campinas, Campinas, Brazil, in 1997, all in electrical engineering. He is currently an Associate Professor with the Department of Electronic Engineering, Federal University of Minas Gerais. His research interests include computational intelligence and fault detection in dynamic systems.